**Paper 501**

# How Will You Use CDISC Library?

Sally Cassells, CDISC, Austin, TX USA

## ABSTRACT

CDISC Library has become the 'true source' of metadata for published CDISC standards. Although many of us have been wishing for this for many years, it can be hard to know how to start using it. Can CDISC Library provide metadata for a study? How can you integrate CDISC Library with your in-house MDR? Is it a tool only for vendors? How can you use CDISC library with your existing CDISC tools? Can CDISC Library drive changes to the standards development process and improve the utility of machine readable metadata? This talk will provide an example of a tool to assess the impact of a new version of a CDISC ADaM Standard and show some ways to use the RESTFUL API to integrate CDISC Library content into your programming environment.

## INTRODUCTION

CDISC Library was rolled out to CDISC members in April of this year (2019). The new technology provided by Nurocor, reworked API and recently announced Data Standards Browser mean that beyond the new name, CDISC Library usability and functionality goes beyond what was possible with CDISC Share. CDISC Library now offers a single 'true source' for the CDISC SDTM, ADaM and CDASH standards. Ironically, now that we are at this point, it is not clear how the CDISC community will use the library. It has been noted that beyond accessing the spreadsheet extracts from CDISC SHARE from the CDISC Members Only site, for a 'typical' SAS programmer the library may not seem very accessible.

A partial list of CDISC Library Use Cases includes

- Viewing contents of published versions of CDASH, SDTM and ADaM Implementation Guide metadata.
- Obtain a spreadsheet with the contents of a published version of CDASH, SDTM and ADaM
- Obtain starting point metadata for CDASH, SDTM and ADaM for the study setup process.
- Assess the impact of a new version of a CDISC content standard
- Integrate CDISC Library with your in-house Metadata Repository
- Using CDISC Library in the Standards Development Process

This paper will provide information for programmers who are interested in exploring CDISC library to investigate how their organization may be able to use to improve its CDISC Standards Implementation capabilities.

## VIEWING CDISC STANDARDS USING THE DATA STANDARD BROWSER

There are probably some programmers who carry detailed knowledge of several CDISC standards versions in their brains 24x7 but for the rest of us, the Data Standard Browser provided as part of the Q3 release of the CDISC Library is a wonderful new tool. Not only does it have an intuitive and appealing user interface but it provides the ability to view the metadata for published versions of CDASH, SDTM and ADaM standards.

If you have a CDISC Library account, the browser is accessible at https://library.cdisc.org/api/browser. The first time you access it, you will need to provide your library username and password but keeping the browser open does not consume your local computer's resources.

## THREE WAYS TO GET A SPREADSHEET

For many users of CDISC standards, the most convenient way to view or use the CDISC standard metadata is with an Excel spreadsheet. There are at least 3 ways, without writing any code, to get a spreadsheet that reflects the complete set of metadata contents for a published standard.

### METHOD 1 – MEMBERS-ONLY LIBRARY ARCHIVE WEB SITE

As has been the practice since CDISC SHARE first became available, the CDISC Members Only web site provides a CDISC Library Archive page (https://cdisc.org/members-only/cdisc-library-archives) where you can download an Excel spreadsheet for each of the published CDISC standards. The set of CDISC Standards and Terminology that

are available through the Library is shown in Figure1 below.

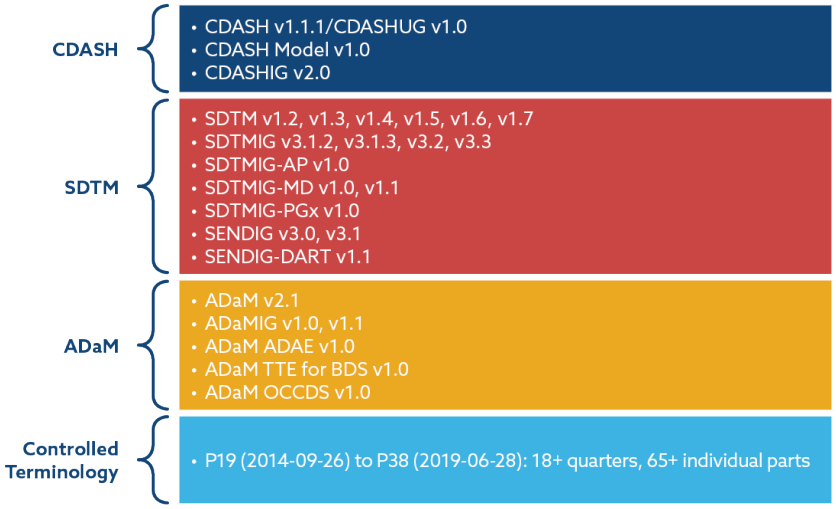


**Figure 1 Published Standards Available in CDISC Library**

**METHOD 2 – DATA STANDARDS BROWSER –TOP LEVEL VIEW**

In the top level view for each CDASH, SDTM or ADaM standard in the library, there is an Export button. Clicking the button allows you to select one of two excel based formats – Comma Separated Values (CSV) or Microsoft® Excel (XLSX). The Excel export will have exactly the same contents as the spreadsheet published in the members-only archive for the respective standard. The CSV export will include only the Variables sheet. Figure 2 illustrates the top-page Data Standards Browser view for the ADaM standards.

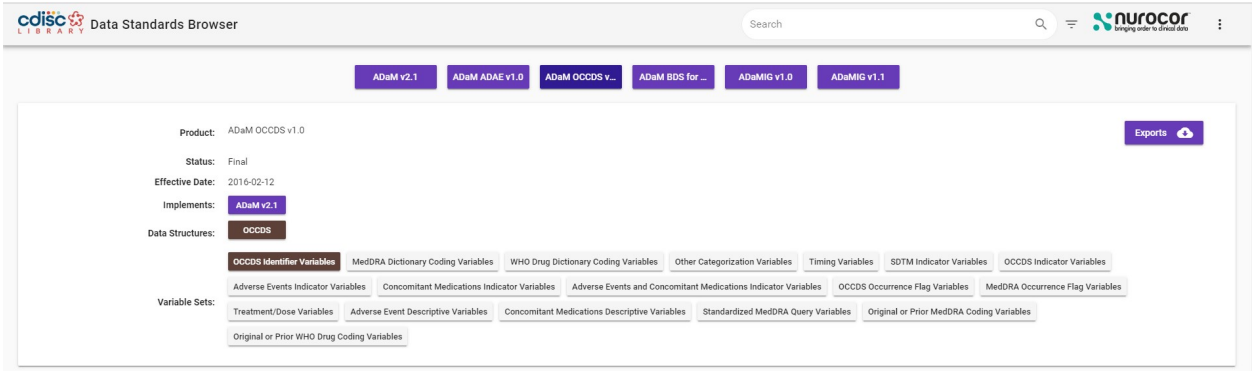


**Figure 2 Requesting Spreadsheet from the Data Standards Browser**

**METHOD 3 POSTMAN INTERFACE**

One way to get started using the Library API is to use an application such as Postman. Postman provides an interactive environment for composing and executing API Requests. There is a short "Getting Started" video demonstration for the Postman interface available at https://wiki.cdisc.org/display/LIBSUPRT/How-to+articles.

The API itself is a set of Restful Web Services endpoints known as requests. Just as there is a top-level view for each standard in the Data Standards Browser, there is a top level request for each standard in the API. A list of all of the requests is available on the CDISC Library Documentation site at https://www.cdisc.org/cdisc-library/api-documentation.

The top-level requests for the most recent versions of the CDASHIG, SDTMIG and the ADaMIG are shown in the table below

**Table 1 Top-level API Requests for 2019 Production Standards Versions**

| Standard Version | Get Request |
|---|---|
| CDASHIG 1.1 | /mdr/cdashig/1-1 |
| SDTMIG 3.3 | /mdr/sdtmig/3-3 |
| SENDIG 3.1 | /mdr/sendig/3-1 |
| ADaMIG 1.1 | /mdr/adam/adamig-1-1 |

Most API requests provide the response in Json, XML or CSV format but for the top level requests, the Excel option is supported. Figure 3 illustrates how to set the API header key to specify that the response should be provided in Excel format. The resulting response payload will not be intelligible but can be downloaded into a local file with an **xls** or **xlsx** suffix. Using this header key option for requests at lower levels of the standard's metadata hierarchy will return an error response.

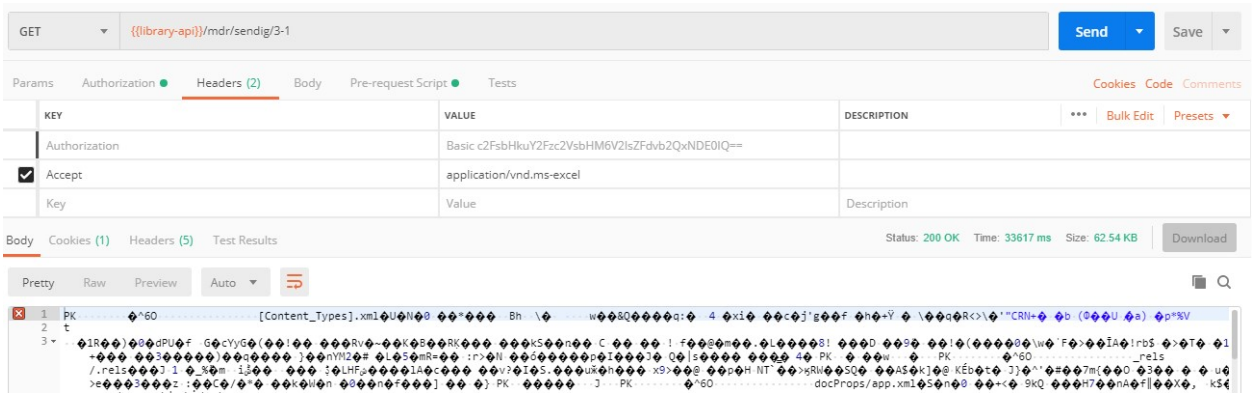


**Figure 3 Setting Postman Request Header for Excel Response**

## USING CDISC LIBRARY STANDARDS AS THE STARTING POINT FO A STUDY

A study protocol defines the activities, assessments and outcomes that will be evaluated as part of a clinical trial. These study specific requirements supplement the legal requirements defined by the government authorities responsible for overseeing development of bio-pharmaceutical products and medical devices. Although the CDISC Library does not yet include the metadata corresponding to the Therapeutic Area User Guide (TAUG) publications, the scope of the developed standards is now sufficiently wide that almost any data you might want to collect in a clinical trial can be mapped into the one of the 60 domain datasets in version 3.3 of the SDTMIG.

The Data Standards Browser provides a good place to begin to explore which SDTM and CDASH domains will be used to manage the data for the assessments and support the study analysis requirements. The set of requests available for each standard reflect that standard's metadata hierarchy. For SDTMIG and SENDID standards, the hierarchy is Class - Dataset – Variable. For CDASHIG the metadata hierarchy is more complex. In addition to the Class/Domain/ Field branch, there is a Class/Scenario/Field Branch. For ADaMIG, the metadata hierarchy is Data Structure/VariableSets/Variables.

The API can then be used to:

- Extract the variable level metadata for a set of SDTM domains then use the extracted metadata to create a Define-XML that will serve as a specification.
- Extract the metadata for a set of CDASH domains. The extracted metadata can be used to create an ODM document to serve as a CRF specification.
- Extract the metadata for a set of CDASH scenarios to assess which scenario(s) will apply best in the clinical setting required for the study assessments.
- Extract the metadata for the ADaM variable sets that will be used to create the ADaM datasets to support the study outcome analysis.

These use cases all require some programming, either to process a series of API requests or to transform the request response into a format that allows for review by a person or an application. There is a set of CDISC Library How-To pages that includes Getting Started: Programmatically connect to CDISC Library which has a set of code snippets that show how to submit a basic request from Python, SAS and Java. While the API can be using within

almost any programming language and provides responses in a variety of formats, experience has shown it is useful to learn a bit about both Python and JSON.

**PYTHON**

Python is a programming language that notable in that it is easy to learn, developed and supported by an Open Source community and used extensively in real-world business applications around the world[3]. A Beginner's Guide to Python at https://wiki.python.org/moin/BeginnersGuide/Programmers provide information about how to download and install a Python interpreter and suggestions for how to go about learning the language.

**JSON**

JavaScript Object Notation is an open standard file format that uses human-readable text to transmit data objects consisting of attribute-value pairs and arrays. The model-responses for the CDISC Library endpoints are provided as JSON code

The code below shows the model response for the request for a the list of CDASH Domains /mdr/cdash/{versions}/domains. Attributes are separated with a comma with the comma separator used for attribute-value pairs. In the example shown below, the top-level attributes are "name", "label", "description", "source", "effectiveData", "registrationStatus", "version","_links" and "domains". Nested attributes, such as "_links", are enclosed with curly braces. List attribute values, such as "domains", are enclosed in parentheses

```
{
 "name": "string",
 "label": "string",
 "description": "string",
 "source": "string",
 "effectiveDate": "2019-09-24",
 "registrationStatus": "string",
 "version": "string",
 "_links": {
  "self": {
   "href": "string",
   "title": "string",
   "type": "string"
  },
  "priorVersion": {
   "href": "string",
   "title": "string",
   "type": "string"
  },
  "domains": [
   {
    "href": "string",
    "title": "string",
     "type": "string"
   }
  ]
 }
}
```

Manipulating JSON structures in Python code is simpler and more intuitive than working with XML and XSL stylesheets.

**CODE EXAMPLES**

Two annotated examples of Python programs that access the CDISC Library API are available in GitHub at github.com/scassells/cdisclibrarytools.

**ASSESS IMPACT OF NEW STANDARDS' PUBLICATIONS**

As much as standards developers look forward to releasing a new version of a standard, it is clear that the work of developing a standard pales before the work of implementing a standard. While the CDISC Library does not provide enough magic to make all that work disappear, the API can facilitate the process of providing reports that detail the

new and updated metadata components.

For ADaM and the SDTM family of standards, the metadata representation in the library includes links to the library structures between those versions at each level of the metadata hierarchy.

In the case of the SDTMIG for example, there are links to the metadata in the previous version at the version, class, dataset and variable levels.

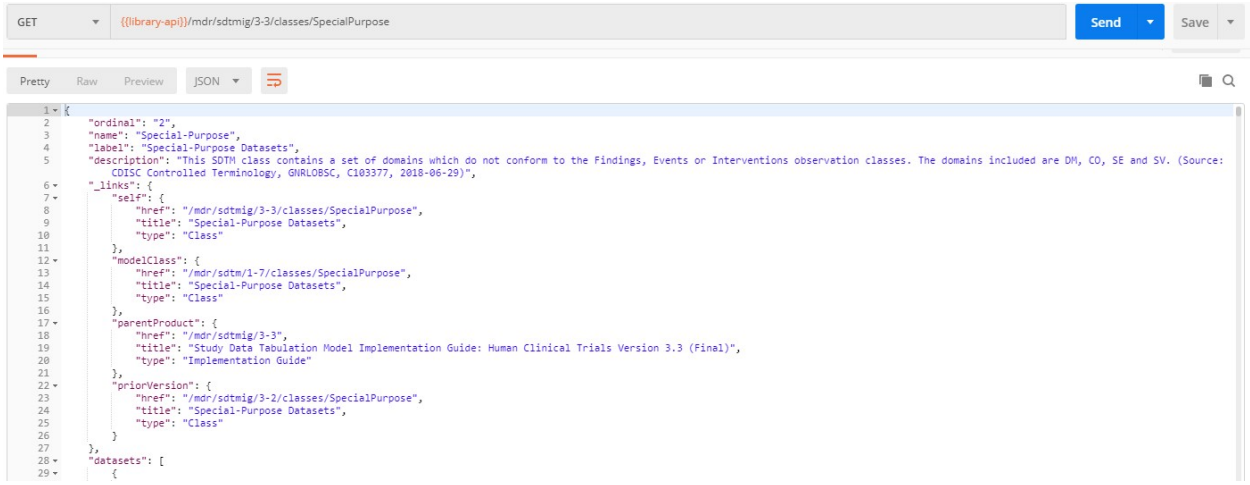


**Figure 4 Link to PriorVersion**

One way to compare two versions of a standard is to create a Define-XML file from the Library Metadata for each standard. The XSL stylesheet display provides a familiar way to navigate through the metadata hierarchy.

Even with the CDISC Library, programmatically identifying all of the changes between two versions of the standards is challenging. There are some inconsistencies in the metadata from version to version, differences in the structures of the metadata hierarchies and differences in how each standards development team documents controlled terminology requirements. Adding built-in API endpoints to provide standardized reports of the differences between each standard and its prior version is on the CDISC Library development roadmap.

In the meantime, one way to compare two versions of a standard is to create a Define-XML file from the Library Metadata for each standard. The XSL stylesheet display provides a familiar way to navigate through the metadata hierarchy.

## INTEGRATION WITH AN IN-HOUSE MDR

For sponsor companies and CROs, implementing an in-house MDR provides a way to govern and manage operational use of CDISC standards. Implementing standards generally involves developing policies and processes for applying the standards within a study. Governance practices are typically designed to ensure consistency across studies, encourage efficient operations and meet regulatory requirements. Operationally an MDR may provide tools for managing internally developed metadata such as code lists designated a sponsor level in the standards, edit checks, form layouts, ETL programs, and statistical analysis tools. It may also reflect decisions about variables that are identified as permissible or optional in the standards publications.

In-house metadata repositories can take many forms. They can be a collection of spreadsheets in Microsoft One-Drive, a bespoke metadata management system backed up by an Oracle database or an off-the-shelf system like Formedix.

At a technical level, the interface between an in-house MDR and the CDISC Library will use the API.The decision about which technical format to use for this interface will depend on the technology used to implement the in-house MDR.

## CONCLUSION

The CDISC Library marks a turning point as it is the first time CDISC has committed to providing a tool to our community with the goal of facilitating the use of new technologies when implementing our standards.

The Library is now the official source of CDISC Standards Metadata but in the future it will also include, in addition, conformance rules and examples. By investing time now to learn to use the CDISC Library API and to build applications in your favorite programming language that allow you make some part of your work easier, you will be ready to take advantage of new features and content as they are made available.

## REFERENCES

1. Stackhouse, Michael & Terek Peterson. “Tame Your SHARE with a PYTHON and SAS” PhUSE US Connect 2019 Poster PP13.
2. https://www.cdisc.org/cdisc-library/api-documentation.
3. https://www.python.org/about/quotes.

## ACKNOWLEDGMENTS

Thanks to Anthony Chow and Julie Chason from the CDISC Data Sciences team.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Sally Cassells (email: scassells@cdisc.org)
CDISC
401 West 15th Street
Suite 800
Austin, TX 78701 USA